

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

1. – 11. (Cancelled)

12. (Currently Amended) A distributed computer system, comprising:
a client comprising a non-cast object graph, wherein the non-cast object graph comprises a non-cast root object and a plurality of non-cast objects;
a server operatively connected to the client;
a client-side transport packager located on the client configured to:
create an internal representation using a variable usage specification, a casting rule,
and the non-cast root object, wherein the variable usage specification lists a
first subset of the plurality of non-cast objects, and
forward the internal representation to the server-side transport packager;
a server-side transport packager located on the server configured to instantiate a cast object graph using the internal representation;
~~means for creating an internal representation using a root object of the object graph;~~
~~means for instantiating a cast object graph using a casting rule and the internal representation;~~
wherein the cast object graph comprises a plurality of cast objects,
wherein each of the plurality of cast objects references at least another one of the plurality of cast objects,
wherein an original name associated with ~~each~~ at least one of the plurality of non-cast objects is modified in accordance with the casting rule, and
wherein the casting rule defines how to modify the original name associated with the at least one each of the plurality of non-cast objects.

13. (Cancelled)

14. (Cancelled)

15. (Currently Amended) The distributed computer system of claim 12, ~~further comprising: wherein creating the internal representation further comprises using means for obtaining a class definition, wherein the class definition corresponds to a template describing methods and at least one selected from the group consisting of variables and constants for one of the plurality of non-cast objects — wherein the class definition is used to create the internal representation.~~
16. (Currently Amended) The distributed computer system of claim 15, wherein the class definition is generated at runtime by a transport packager using one selected from the group consisting of reflection and introspection.
17. (Original) The distributed computer system of claim 12, wherein the casting rule comprises a casting method.
18. (Currently Amended) The distributed computer system of claim 17, wherein the casting method implements a mapping method, wherein the mapping method comprises mapping the original name of one of the plurality of non-cast objects to a new pre-determined name.
19. (Currently Amended) The distributed computer system of claim 17, wherein the casting method implements a suffix method, wherein the suffix method comprises appending a suffix to the original name of one of the plurality of non-cast objects.
20. (Currently Amended) The distributed computer system of claim 17, wherein the casting method implements a parser method, wherein the parser method comprises replacing the original name of one of the plurality of non-cast objects with a name corresponding to a superclass of a class to which the one of the plurality of non-cast objects belongs.
21. (Original) The distributed computer system of claim 12, wherein the internal representation is a serialized file.
22. (Cancelled)
23. (Cancelled)

24. (New) The distributed system of claim 12, wherein each of the first subset of the plurality of non-cast objects in the variable usage specification is defined using a path, wherein the path specifies all non-cast objects in the non-cast object graph between the non-cast root object and the one of the first subset of the plurality of non-cast object.
25. (New) A system, comprising:
a client, operatively connected to a server, comprising a non-cast object graph, wherein the non-cast object graph comprises a non-cast root object and a plurality of non-cast objects and wherein the client is configured to issue a request to invoke a remote method the server; and
a client-side transport packager located on the client configured to:
intercept the request,
create, in response to the request, an internal representation using a variable usage specification, a casting rule, and the non-cast root object, wherein the variable usage specification lists a first subset of the plurality of non-cast objects,
forward the internal representation to the server, and
receive a response to the request from the server,
wherein the server comprises a server-side transport packager configured to instantiate a cast object graph using the internal representation and generate a response the request using the cast object graph,
wherein the cast object graph comprises a plurality of cast objects,
wherein each of the plurality of cast objects references at least another one of the plurality of cast objects,
wherein an original name associated with each of the plurality of non-cast objects is modified in accordance with the casting rule,
wherein the casting rule defines how to modify the original name associated with each of the plurality of non-cast objects, and
wherein each of the plurality of cast objects is configured to store at least one of the plurality of attributes.

26. (New) The system of claim 25, wherein creating the internal representation further comprises using a class definition, wherein the class definition corresponds to a template describing methods and at least one selected from the group consisting of variables and constants for one of the plurality of non-cast objects.
27. (New) The system of claim 26, wherein the class definition is generated at runtime by a transport packager using one selected from the group consisting of reflection and introspection.
28. (New) The system of claim 25, wherein the casting rule comprises a casting method.
29. (New) The system of claim 28, wherein the casting method implements a mapping method and wherein the mapping method comprises mapping the original name of one of the plurality of non-cast objects to a new pre-determined name.
30. (New) The system of claim 28, wherein the casting method implements a suffix method, wherein the suffix method comprises appending a suffix to the original name of one of the plurality of non-cast objects.
31. (New) The system of claim 28, wherein the casting method implements a parser method, wherein the parser method comprises replacing the original name of one of the plurality of non-cast objects with a name corresponding to a superclass of a class to which the one of the plurality of non-cast objects belongs.
32. (New) The system of claim 25, wherein the internal representation is a serialized file.
33. (New) The system of claim 25, wherein each of the first subset of the plurality of non-cast objects in the variable usage specification is defined using a path, wherein the path specifies all non-cast objects in the non-cast object graph between the non-cast root object and the one of the first subset of the plurality of non-cast object.